Function-to-Form Mapping for Tolerance Synthesis: Part-II: Conceptual Design and Tolerance Synthesis

U. Roy[*] and N. Pramanik
Knowledge Based Engineering Laboratory
[*]Dept. of Mech, Aerospace, Manufacturing Engineering
Syracuse University
Syracuse, NY 13244-1240
Email: {uroy, pramanik}@ecs.syr.edu

R. Sudarsan, R. D. Sriram and K. W. Lyons
Engineering Design Technologies Group
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
Email: {sudarsan, sriram,klyons}@cme.nist.gov

Abstract

In this paper, a design synthesis process has been proposed for the evolution of a conceptual design from the product specification and a proactive approach to tolerance synthesis has been proposed in the early stages of design when the product realization process is still evolving. The proposed design synthesis method is a mapping from the functional requirements to artifacts, with multi-stage constrained optimization during stages of design evolution. An overall design scheme has been proposed including optimization of global goals involving manufacturability, assembliability, and cost. Tolerance models for synthesis of tolerance during the detailed design phase has been introduced. The methodology presented in this paper, uses generic definitions of product specification, function requirements, behavioral models, and tolerance models introduced in the Part-I of this work.

1.0 Introduction

Though it is true that the tolerance design is completed (as a full specification of tolerances needed for any assembly) only when the whole assembly is finished and its components are duly detailed, the design for tolerances should be started much earlier in the conceptual phase of the design to direct the search procedure in a large design space. In this paper, we intend to study the role of tolerance design (in order to develop a proper tolerance specification) on the overall function-to-form mapping process in order to realize a quality and cost-effective design solution in the conceptual design phase. We believe that significant gains can be achieved by effectively using tolerancing issues into consideration during the early stages of the design process and both the product structures (form) and its associated tolerance information should evolve continuously from the given functional specifications.

2.0 Design Synthesis

The design of an artifact to satisfy the product specification (PS) is a complicated process. The design process is considered evolutionary in nature [1]. We start with incomplete knowledge to look for suitable artifacts and/or functional entities in the corresponding library to arrive at a design starting point. At this stage, some of the attributes specified in PS may have been found and some of the constraints may have been satisfied. In order to proceed further, more knowledge is required to be injected into the system and the set of specifications are needed to be transformed for subsequent enhancement of the initial solution. Here the design of an artifact is defined as an object with two elements **D ::= {<PS><Art_Tree>}** where <PS> is Product Specification, <Art_Tree> is the artifact tree (a tree structured list of artifacts). A detailed description of the product specification PS and other constructs used in this paper, are available in the part-I [2] of this work where generic definitions for these entities are developed.

Initially, the artifact tree is empty. Subsequently, when suitable artifacts are mapped to perform a desired functionality, these artifacts are added to the artifact tree. Outputs from an artifact that are not in the PS go as input to the next artifacts. Outputs that are found in the PS are terminals. Also, the designer can mark an output as terminal so that further mapping of this output as input to a new artifact is not required. This process develops the artifact tree.

Above approach for design synthesis generates stages of (sequence of) of partial solutions as shown below.

$$\begin{aligned} D_0 &= \{<PS_0><Art\_Tree_0>\} \\ D_1 &= \{<PS_1><Art\_Tree_1>\} \\ D_2 &= \{<PS_2><Art\_Tree_2>\} \\ &\dots \\ D_n &= \{<PS_n><Art\_Tree_n>\} \end{aligned}$$

Where, at the beginning of the design process, $<Art\_Tree_0>$ is NULL.

[*] corresponding author

As would be detailed later in this section, at each stage of the design evolution, the partial solution is checked for convergence to the desired output specified in the PS. This checking is performed using two basic criteria: a constrained norm minimization process involving the relational constraints associated with the product specification as well as the individual artifacts. The norm (defined later in this section) is the 'distance' of the partial solution from the desired output. After the minimization, satisfaction of spatial constraints is checked. Based on the above two criteria, the set of candidate artifacts are graded from 'best' to 'worst' at that particular stage. We have introduced a *design control parameter,* $\mathbf{N_{alt}}$ as the number of artifacts (that are most desirable in the graded list) to be considered for the next stage. This implies that, for example, if in an intermediate stage of the design evolution, 10 artifacts are mapped and N_{alt} has been set by the designer as 3, only the top three of these 10 artifacts will be used as possible candidates in this stage and searching would continue from those 3 artifacts for the the next stage. It may be noted here that as N_{alt} increases, possibilities of more diverse solutions increase, which is a desirable feature, since more alternative design solutions can be explored. However, there is a cost associated with the increase in N_{alt} in terms of computation time and storage requirements. In the proposed system, we have planned to keep this design control parameter N_{alt} as a designer selectable value. A suitable value could be decided by further studying the design synthesis process with different product specifications in a design domain.

The design synthesis process at some intermediate stage will have at most $\mathbf{N_{alt}}$ branches from each of the artifacts in that particular stage. The process of expanding a particular branch will terminate when one of the following conditions has been reached.

i) A feasible solution satisfying the output specification, relational constraints as well as spatial constraints have been satisfied. This means that the minimization process discussed earlier has resulted in an acceptable distance between the desired output in PS and the partial solution. We designate this acceptable distance as a *convergence criterion,* Î $_0$. Thus **d <=** Î $_0$ is the termination criteria.

ii) The search for a suitable artifact from the artifact library failed to map at least one artifact and hence the design synthesis process cannot proceed further. How to proceed with an alternative scheme for further search has been discussed later.

There are some basic considerations in the design evolution process depicted above which need further investigations. These are: Transformation of PS_n to PS_{n+1}, including attribute transformation, constraint transformations, and variation of internal parameters of artifacts for searching a solution as a minimization of the above-mentioned norm (distance between the desired solution and current stage of solution). These have been discussed in the following sections.

2.1 Product Specification (PS) Transformations

In this subsection, we discuss the details of product specification transformations, which are required at each stage of the design synthesis process. The Product Specification transformation consists of Attribute Transformations, Constraint Transformations and the Variation of internal parameters. These have been discussed in the following sections.

2.1.1 Attribute Transformation

The product specification $\mathbf{PS_0}$ contains the initial specification with $\mathbf{PS_0.Inp}$ and $\mathbf{PS_0.Out}$ as the sets of input and output specifications, respectively. Assuming that at stage **j,** a sub-set of these sets of requirements have been satisfied, $\mathbf{PS_j}$ is transformed into $\mathbf{PS_{j+1}}$ as described below.

Let us assume that an artifact, $\mathbf{Art_{jk}}$ has been found in the design stage **j** with some elements of $\mathbf{Art_{jk}.Inp}$ are in $\mathbf{PS_j.Inp}$ and some elements of $\mathbf{Art_{jk}.Out}$ are in $\mathbf{PS_j.Out}$. We can represent this as union of two mutually exclusive sets:

$\mathbf{Art_{jk}.Inp = Art_{jk}.Inp_1}$ È $\mathbf{Art_{jk}.Inp_2}$
$\mathbf{Art_{jk}.Out = Art_{jk}.Out_1}$ È $\mathbf{Art_{jk}.Out_2}$

where $\mathbf{Art_{jk}.Inp_1}$ Í $\mathbf{PS_j.Inp}$ and $\mathbf{Art_{jk}.Inp_2}$ Ë $\mathbf{PS_j.Inp}$
and $\mathbf{Art_{jk}.Out_1}$ Í $\mathbf{PS_j.Out}$ and $\mathbf{Art_{jk}.Out_2}$ Ë $\mathbf{PS_j.Out}$

If $\mathbf{Art_{jk}.Inp_2}$ is NULL then all input requirements of the artifact $\mathbf{Art_{jk}}$ are in the product specification $\mathbf{PS_j.Inp}$ and this artifact needs no further artifacts whose output should be mapped to inputs. Otherwise, the inputs need to be transformed in to a new set of outputs specifications for some artifact to be searched with:

$\mathbf{PS_{j+1}.Out = PS_j.Inp}$ È $\mathbf{Art_{jk}.Inp_2}$

If $\mathbf{Art_{jk}.Out_2}$ is NULL then all outputs of the artifact $\mathbf{Art_{jk}}$ are in the product specification $\mathbf{PS_j.Out}$ and the outputs of this artifact need not be mapped as input to some other artifact. Otherwise, the outputs need to be transformed to a new set of inputs for some artifacts. The designer, if desired, can accept some of these outputs as byproducts to the environment and treat them as already satisfied. The remaining outputs are then transformed into a set of new input specification as:

$\mathbf{PS_{j+1}.Inp = PS_j.Out}$ È $\mathbf{Art_{jk}.Out_2}$

2.1.2 Constraint Transformation

Constraints play a major role in any design by restricting the design space from an open-ended search to a more restrictive (and hopefully, of polynomial time) search. In other words, constraints could be thought of as a guiding mechanism for evolving a design along some restricted path.

In this work, constraints have been categorized into two separate groups for ease of treatment/management. These are relational constraints, and spatial constraints.

Relational constrains are direct functions of attributes (or parameters of attributes) according to some physical law or some other restrictions.

<relational>::=**f**(<attribute_name>[,<attribute_name>]…)
EQ <value_range>, where <value_range> is an interval for the possible values of the function. It has been mentioned during discussion on artifact representation that in general, the value of any parameter (an attribute) could be in a range (closed interval).

The function **f** could be of three types: explicit, implicit or parametric.
<explicit|implicit> ::= $f(X) \in \mathbf{R}$, $X \in \mathbf{R^n}$
<parametric> ::= $f(X(t)) \in \mathbf{R}$, $t \in \mathbf{R^n}$: $t_j \in (0,1)$ &
$X_j = X_{j0} + t_{j*} (X_{j1} - X_{j0})$

If **f** is a vector valued function, it would be treated as a set $(f_1,f_{2,\ldots}f_n)$ of *n* scalar functions such that $f_j \in \mathbf{R}$, $j\in(1,n)$

For example, in a rotary motion transformation, a global constraint requiring a speed ratio (assuming ω_I as input and ω_O as output rotary speed) could be:

PS. ω_I.value/PS. ω_O.value EQ (5,6); a reduction of 5 to 6.

Constraints Transformation

Constraints associated with parameters of an artifact have to be satisfied. This would be a straightforward process of applying the available range of values for the parameters to check the constraint equations whether the set of values satisfy or fail to satisfy a constraint. However, in general, some of the constraints may not be fully satisfied and in such cases, the effect of the constraint should be transferred to the next artifact. This is called the constraint transformation and propagation. The procedure for such transformation is as follows:

A constraint of the form $f(y_1,y_2, y_3, \ldots y_n) = 0$ would be converted to a set of n equations, by solving for each y_j in terms of the others.

$$y_1 = f_1(y_2, y_3, y_4 \ldots)$$
$$y_2 = f_1(y_1, y_3, y_4 \ldots)$$
$$y_3 = f_1(y_1, y_2, y_4 \ldots)$$
….

If such an explicit representation is not possible, the constraint may have to be represented in a different way, either by linearizing, or by approximating into simpler forms.

If an attribute from the artifact is linked to another artifact, two possible cases are there: an output attribute goes as an input to the next artifact or an input attribute comes out as an output. In either case, we use the corresponding component of the constraint and solve for the new range for the parameter. This new range accompanies the attribute as a constraint to the next artifact.

In the next artifact, there may be a priori knowledge about the range of that attribute within which that artifact operates. In order to check that the incoming attribute value range is acceptable, an intersect of the two intervals are performed as: P_{in} Ç $P_{allowable.}$ If the intersect is NULL, there is a contradiction and the constraints associated with the incoming attribute P makes the new artifact unsuitable for a possible element of the artifact tree.

Spatial Constraints

These constraints are relations amongst attributes linking forms /geometry of the artifacts. These would represent spatial (structural) relationship between attributes having shape / size / orientation related properties. Following is a generic definition for these constraints:

<spatial> ::= <attribute> <spacial_relationship>
[<attribute>] [a_value]
<spatial_relationship> ::= <orientation> <position>
<connection>
<orientation> ::= <direction cosine of major axis of
attribute1 w.r.t. that of some attribute2> .
<position> ::= co-ordinate of center of attribute1 w.r.t.
center of attribute2
<connection> ::= <connection_type><contact_details>
<connection_type> ::= <point2point| point2surface|
surface2surface| etc…>
<contact_details> ::= <set of points, surface, and common
dof of connection.>

Some common orientations are: horizontal, vertical, perpendicular_to, parallel_to, distance_from, etc. As for example, (for a chair), we might have following spatial constraints:

('arm' parallel_to 'seat')
('backrest' perpendicular_to 'seat')
('seat' horizontal_to 'base')
('seat' distance_from 'base' 2 ft)
….

2.1.3 Variation of Internal Parameters of Artifacts for Selecting an Artifact

As it has been pointed out in earlier discussion in this paper, artifacts are searched from the artifact library by matching input parameter types for possible candidates in the solution. However, a suitable measuring and optimizing criteria would be required for guiding the solution. In other words, some criteria for selecting the 'best' possible candidate at each stage from a possible set of artifacts have to be formulated.

We define a 'distance' type norm for measuring the proximity between the desired output (as specified in PS) and the partial solution reached at some stage j, as:

$d(a,b) = ((a_{low}-b_{low})^2 + (a_{high}-b_{high})^2)^{1/2}$
where a and b are two variables representing intervals $a = (a_{low},a_{high})$ and $b = (b_{low},b_{high})$

Above definition satisfies properties of a norm:

$d(a,b)=0$ iff $a_{low} = b_{low}$ and $a_{high} = b_{high}$
$d(a,b)>0$ for a != b
$d(a,b) = d(b,a)$

We sometimes would use a parametric form to represent intervals a and b.
As for example, $a = a_{low}+(a_{high}-a_{low})*\theta : \theta \in (0,1)$

While the range of feasible variations of the input will be used to check for suitability of accepting an artifact, the variations allowed in the internal parameters of an artifact would be used to minimize the 'distance' between the desired output (as specified in PS) and the output (partial solution) at an intermediate stage. The minimization scheme is formulated as below:

Minimize: $d(O_j, O_0)$, where, O_0 is the output specified in the PS and O_j is the output from the artifact j in an intermediate stage of the design.

The partial solution O_j is given by: $O_j = f_j (I_j, \beta_j)$, which is derived from the main constraint C_0 (relationship between the input and the output of the artifact j, [1]), by solving for O_j from $C_{j0} (I_j, O_j , \beta_j) = 0$.

The parameter β (where $\beta_j = (\beta_{j1,} \beta_{j2,\ldots} \beta_{jn})$) is the internal parameter of artifact j. The parameter β expressed in parametric form would be:
$\beta_{jk} = \beta_{jk\_Low} +(\beta_{jk\_High} - \beta_{jk\_Low})* \theta_{jk} : \theta_{jk} \in (0,1), k \in (0, n)$

The subscripts Low and High indicate the lower and upper bounds of the interval for β_{jk}.

It is also possible that apart from the C_0 constraint, an artifact may have additional relational constraints associated with it. These relational constraints are expressed as: $C_k(I, O, \beta)$ <rel_opr> <value> , k>0 and <rel_opr> is the relational operator (one of {LT LE EQ GE GT NE}), and <val> is a numeric value range. For the optimization scheme, these relationships are converted to the standard equality form $C_k(I, O, \beta) = 0$, by introducing additional variables for the cases where the <rel_opr> is not "EQ").

The input to the artifact *j*, I_j is equal to the output from the previous artifact *j-1* and so on. These gives rise to the chain of linked equations and the optimization scheme becomes:

Minimize: $d(O_j, O_0)$ subject to:
; constraints associated with artifact j
$I_j = O_{j-1}, C_{j,1}(I_j, O_j , \beta_j) = 0, \quad C_{j,1} (I_j, O_j, \beta_j) = 0, \ldots,$
$C_{j,c(j)} (I_j, O_j, \beta_j) = 0$
; constraints associated with artifact j-1
$I_{j-1} = O_{j-2}, C_{j-1,0} (I_{j-1}, O_{j-1,} \beta_{j-1}) = 0, C_{j-1,1} (I_{j-1}, O_{j-1}, \beta_{j-1})=0$
$, \ldots, C_{j-1,c(j-1)} (I_{j-1}, O_{j-1}, \beta_{j-1}) = 0$
; constraints associated with artifact j-2
$I_{j-2} = O_{j-3}, C_{j-2,0} (I_{j-2}, ,O_{j-2} , \beta_{j-1}) = 0, C_{j-2,1} (I_{j-2}, O_{j-2}, \beta_{j-2}) = 0,$
$\ldots, C_{j-2,c(j-2)} (I_{j-2}, O_{j-2}, \beta_{j-2}) = 0$
…
; constraints associated with artifact 1
$I_2 = O_1 , C_{1,0} (I_1, O_1, \beta_1) = 0, C_{1,1} (I_1, O_1, \beta_j) = 0, \ldots,$
$C_{1, c(1)} (I_1, O_1, \beta_1) = 0$

where '1+c(k)' is the number of constraints associated with artifact k.

Above minimization scheme could be solved using Lagrange multiplier scheme by including the constraints into the main optimization function as:

$$\boldsymbol{d_j = d(O_j , O_0) + S_{n \hat{I} (1,j)} S_{k \hat{I} (0, c(n))} (m_{n,k} * C_{n,k}(I_n , O_n , b_n)) + S_{p \hat{I} (1,j-1)} (l_p * (I_{p+1} - O_p))}$$

where 1+c(n) is the number of constraints associated with artifact n, and μ's and λ's are Lagrange multipliers.

The minimization of d_j produces a set of parameters (β^*_n), for each artifact n ($n \in (1 , j)$), which makes the present solution closest to the desired solution. We denote by d_j^* and O_j* the corresponding optimal distance and solution. If the value of $d_j*(O_j*, O_0)$ is within a specified value $\in_0$ (convergence criterion), we can accept the current design solution given by D_j = {<PS$_j$><Art_Tree$_j$>} as a feasible solution. However, if the distance d_j* is not within acceptable limit, the solution at this stage represents a partial (an incomplete) solution i.e. the desired output value has not yet been achieved yet .

The above minimization process deals with the relational constraints only. After the above minimization has been performed, (irrespective of the solution whether an acceptable feasible solution or a partial solution), the spatial constraints are then checked for. There can arise four situations after the spatial constraints are applied.

i) A feasible solution has been achieved and the spatial constraints are all satisfied.

ii) A feasible solution has been achieved and all the spatial constraints are not satisfied.

iii) An incomplete solution has been achieved and the spatial constraints are all satisfied.

iv) An incomplete solution has been achieved and all the spatial constraints are not satisfied.

Case i) represents a complete solution and the corresponding branch of the tree can be terminated without further growth. The rest three cases are incomplete and the branching / growth of the solution tree continues to the next stage.

2.2 Design Synthesis Process

The basic procedure for the proposed design synthesis is as follows:

Develop design domain specific artifact library (ARTL), functional equivalence library (FUNL) and domain knowledge base (DK). For the time being, we assume that the DK is specified in the form of constraints and relations in the PS itself. However, these could be separated out for treating them in a generic way.

0. Start with a product specification PS.

1. Locate suitable artifacts from the ARTL mapping the input parameters from the product specification with those of the artifacts having same input type. If no artifacts are found, go to step 8.

2. Check whether the type of output from some of these artifacts matches the output types specified in PS. Divide the artifacts into two sub-groups: one with artifacts whose output matches the desired output (D_M) and the other one where such a match is not found (D_{NM}).

3. With the group D_M

   Generate the distance function between the output and the desired output in PS and minimize the distance along with the constraints associated with the attributes.

   a) If the distance for some of the artifacts is within a specified acceptable value, a possible solution has been found.
   b) Apply the spatial constraints to these artifacts. If these constraints are satisfied, go to step 9.
   c) If the distance is not within the acceptable value, only a partial solution has been found. Take the top Nalt artifacts nearest to the solution. Go to step 4.

   With D_{NM}

   In this case, the minimization criteria can not be applied yet since the output type did not match the desired output type in the PS. The minimization scheme can only be applied when a match for the desired output has been found.

4. Generate new sets of attributes and transform the constraints to augment the PS so that additional attributes associated with the selected artifacts could be taken into account.

5. Repeat steps 1 to 4 with transformation of the product spec PS.

6. Continue till such time all the attribute requirements are satisfied or some attributes could not be mapped.

7. At any stage, if some attributes could not be mapped, there would be three alternatives: look for a possible functional equivalence class and modify the PS accordingly and continue search. If such a functional equivalence class is not found, consult with the designer to acquire new attributes, knowledge, constraints and/or modify existing specification. Repeat steps 1-4 after such modifications. If above steps still fail to map some attribute requirements, the designer needs to add new artifacts in ARTL and/or add new functional equivalence classes in FUNL. After this step, repeat again.

8. In case all options have been exhausted at an intermediate stage, consider the possibility of going back one step and consider other paths with artifacts with lesser matches.

9. After a feasible solution has been found, a tentative sizing of the components of the artifacts is carried out by using the attribute values specified and by applying the physical laws governing the behavior of the artifact. If during this process, some parts could not be sized within acceptable range of values, consider possible change of the PS and go to step 7.

10. Introduce tolerance models associated with each artifact in the artifact tree and carry out tolerance analysis. If during this process, tolerance requirements for some parts are not feasible, consider changing PS and go to step 7.

11. Consider manufacturability of the artifacts in the design solution. Apply criteria for manufacturability. If during this process, some manufacturing requirements for some parts are not feasible, consider changing PS and go to step 7.

12. Consider global goals and constraints associated with the product specification. If the global constraints are satisfied, initiate global optimization processes and consider changing the PS again to achieve some global goals and go to step 7.

13. A feasible design has been arrived at.

The iterative design synthesis process will terminate when one of the followings is satisfied.

1. All the attributes in the PS has been found and the desired output value level has been achieved. In this case, a feasible solution has been found. Now, the global constraints and goals could be evaluated.

2. Some of the attributes are yet to be found and no further artifact could be located in ARTL. In this case, either the designer will provide some more domain specific knowledge in the PS_n or some new artifacts would be added to proceed further. However, in order to explore other possible solutions, we may backtrack one step to D_{n-1} and consider other less favorable possibilities.

Observations on the design synthesis process

In general, an artifact may have more than one input and output attributes and constraints associated with them. In order to consider the artifact as an element of the solution, these attributes are also to be considered as part of the design specification. Thus, we need to augment the design specification with the unsatisfied attributes of this artifact.

If some of the input and output attributes are already in PS, we mark them as found the remaining attributes need to be satisfied. Since these inputs and outputs were not in the original product specification, they are not desirable from the product specification requirement. However, these must be mapped to other artifacts. We would put a negative weight to these attributes (undesirable?) and augment the PS with these new sets of attributes along with associated constraints.

With this augmented PS, we will now search for artifacts from the ARTL. The input attributes must come as output from some other artifact or from a *terminal* that we also consider as artifact with no input and one output (like an electricity supply point as a terminal that supplies electric energy and need no further input)

The output attributes must either be accepted as an undesirable byproduct to the environment and no further exploration would be required or the output must be mapped as in input to some other artifact.

The minimization process mentioned in step 3 above assigns optimum values for the internal parameters of each artifact. After the minimization, if the distance d is within a specified value of $\in_0$, the solution has converged to a feasible solution. However, if d is still not within the range, we continue to add another possible chain of artifacts, and optimize. The process repeats till the desired level has been reached.

2.3 Overall Scheme of the Proposed Design Synthesis Process

Overall scheme of operation for the proposed conceptual design to tolerance synthesis model would be as below:

Step #0 Develop design domain specific Artifact Library, Function Library and Knowledge Base.
While the above three would be represented as objects in the core module (**C++**), artifacts in the Artifact Library will also have references to corresponding **CSG/BREP** representations as **Pro/E** assemblies/parts.

Step#1 Develop product specification based on customers specification.

Step#2 Execute the main design decomposition process in the core module (**C++**).

Step#3 Carry out artifact behavioral study (using behavioral simulation tools).

Step#4 Carry out preliminary dimensional sizing using inherent physical law and specified requirements.

Step#5 Introduce tolerance models associated with each artifact and carry out tolerance analysis.

Step#6 Carry out manufacturability studies.

Step #7 Carry out overall goal analysis.

3. Kinematic Behavioral Model and Tolerance Synthesis

For tolerance synthesis and analysis, we principally need a detailed description of the “kinematic functions” of the assembly, by which we mean those functions defined essentially by the location, size and shape (form) of associated mating features. These are the functions which the geometric dimensioning and tolerancing scheme is primarily concerned to maintain. However, these kinematic functional specifications are not directly provided by the customer’s need statements or by early specifications of the desired product/assembly function. They are slowly evolved with the assembly as the later takes concrete shape and size in the later phases of the conceptual design. Tolerance synthesis and analysis needs an exhaustive functional (kinematic) analysis mechanism to make sure that the identified functional requirements between the mating components of the assembly are met and are suitably described typically in the form of critical toleranced dimensions/size/sizes/forms or in the form of toleranced gaps.

The kinematic behavior model (KBM) is appropriate only at the functional face level. It should be deduced from the part's (or assemblies) structural behavioral model. The first step in this process is to assemble a qualitative model (possibly as a set of qualitative differential equations) of the part/assembly's operation. The qualitative model provides a good knowledge of functional relationships that exist between the parts of an assembly. This model will be further used to identify the functional faces on the part of the assembly (mainly the contact/mating surfaces) and any related functional face assemblies. A "functional relationship graph (FRG) [3, 4]" is then established from the causal dependency graph. This FRG will clearly establish the kinematic functional and behavioral relationships between the mating parts of an assembly at their respective functional faces.

Finally, the kinematic model of each part in the assembly is derived. In order to incorporate the behavioral aspects, a part is then described by its bounding faces and each face is represented as a set of seven (7) tuples $\{K_N, K_T, K_L, F_N, F_L, F_T, P_{behavior}\}$, where K & T represent kinematic and force degrees of freedom respectively along the normal, transverse and longitudinal axes. The $P_{behavior}$ term represents the behavioral attributes (magnitude and direction) of the part behavior (e.g. contact pressure, rotational speed, linear velocity, etc). The kinematic dofs represent the presence/absence of constraints for motion along a particular axis. A combination of two kinematic dofs can be used to represent rotational motion and constraints imposed on the movement of a face about any one of the above three axes. For more information on the KBM, please refer to [5].

Tolerance Synthesis

In order to synthesize tolerance, we follow the procedure suggested by Roy and Bharadwaj [6]. The conceptual schema for tolerance synthesis is shown in figure 1 [6]. Given the design function requirements, manufacturing processing information and assembly plan, the schema helps assign both dimensional and geometric tolerances (along with required datum reference planes) to be part of an assembly.

The tolerance synthesis schema starts with collecting the following information from the aggregate function_behavior_assembly data model (please refer to [1,7] that has been evolved during the conceptual design synthesis process. Following four types of information are necessary:

1) Geometry description:

   assembly level - position and orientation information for each component artifact within the assembly.

   part level - spatial location of form features in the component artifact and their inter-relationships.

   feature level - feature geometry.

2) Functional and Behavioral Specification:

3) Material and Surface finish Specifications:
   Material and surface characteristics should be either retrieved from the database or supplied manually by the user.

4) Assembly graph:
   The procedure for assembling different component artifacts in the assembly (without considering the effect of tolerances) should be retrieved from the data model.

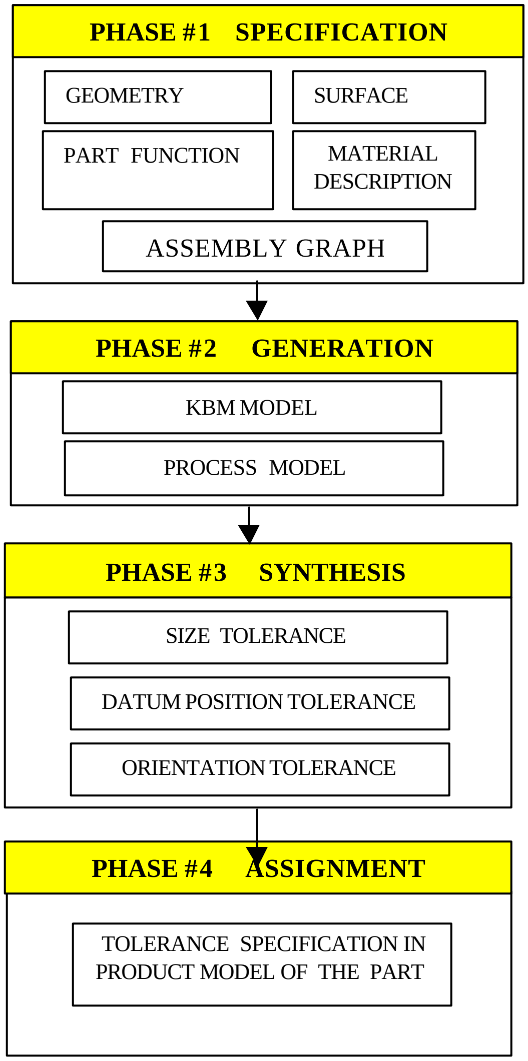

Figure 1. Tolerance Synthesis Scheme [6]

In the second phase, the kinematic behavioral model (KBM) and the process model for each component of the assembly are generated. A kinematic behavioral model describes the spatial and design relationships that exists on the mating faces of a part in terms of certain kinematics and force degrees of freedom (dofs) presence/absence of motions and transmission of forces along the particular axes of a surface. The process model represents the process plan for manufacturing the part without considering the effect of tolerances [8].

The third phase of the schema is the synthesis stage. Different types of tolerances are synthesized for each part of the assembly. It consists of two major tasks: (i) transformation of the KBM model into functional tolerance limits, and (ii) constraining the functional tolerance limits with respect to different manufacturability and assembliability constraints. The first task can be achieved by developing appropriate application domain-specific KBM-to-Functional-Tolerance-Limit maps (refer to [7,9] for a detailed discussion); and the second task can be achieved by developing optimization problems which contain both the functional tolerance limits and the different constraints.

In the fourth phase, dimensional and geometric tolerances (along with the datum specifications) are fine-tuned with respect to the design functions and manufacturing constraints.

4. Conclusion

In this work, we have proposed a design synthesis methodology (with an object-oriented generic approach for function-to-form mapping) for design of products using the representational schemes of product specification, functional requirements, artifact representation, and tolerance representation as described in part-I [2]. However, there are two important aspects of the proposed system, which need further work/research:

i) Detailed study of artifact functional behavior (both qualitative and quantitative) as well as kinematic behavior using suitable behavior modeling tools.

ii) Schemes for optimization of global goals associated with the final product (including manufacturability, assembliability and tolerances) to further improve the design.

The main emphasize of this work has been the study of function-to-form mapping in the product development context as well as the integration of tolerancing schemes in the design process at an earlier stage. Large scale assembly issues, including the intricate problem of evolving both the assembly structure and its associated tolerance information simultaneously needs to be addressed in future.

5.0 Acknowledgments

This work is sponsored by the SIMA (Systems Integration for Manufacturing Applications) program in NIST and the RaDEO program at DARPA.

6.0 References

[1] U. Roy, R. Sudarsan, Y. Narahari, R. D. Sriram, K. W. Lyons, M. R. Duffey, and N. Pramanik. *Information Models for Design Tolerancing: From Conceptual to the Detail Design.* Technical Report, National Institute of Standards and Technology, 1999.

[2] U. Roy, N. Pramanik, R. Sudarsan, R. D. Ram, and K. W. Lyons. Function-to-Form Mapping for Tolerance Synthesis: Part-I: Model and Representation. Submitted for publication in the ASME 2000 IDETC/CIE 20th Computers and Information in Engineering (CIE) Conference, Baltimore, Maryland, September 10-13, 2000.

[3] U. Roy, P. Banerjee, and C. R. Liu. *Design of an Automated Assembly Environment.* Computer-Aided Design, Vol. 21 (1989), No. 9, pp. 561-569 (also published in Robotics, Automation and Management in Manufacturing Bulletin, Vol. 7, Issue 1, Jan. 1990).

[4] Utpal Roy, and C. R. Liu. Establishment of Functional Relationships between the Product Components in Assembly Data Base. J. Computer-Aided Design, Vol. 20 (1988), No. 10, pp. 570-580.

[5] U. Roy, and B. Bharadwaj, Design with Part Behaviors: Behavior Model, Representation and Application. Journal of Computer-Aided Design (in press).

[6] Utpal Roy, and Balaji Bharadwaj. *Tolerance Synthesis in a Product Design System.* Technical Paper# MS96-146. North American Manufacturing Research Institution. Society of Manufacturing Engineers, Dearborn, MI, 1996.

[7] U. Roy, R. Sudarsan, R. D. Sriram, K. W. Lyons, and M. R. Duffey, “Information Architecture for Design Tolerancing: from Conceptual to the Detail Design,” accepted for presentation and publication in the Proc. of DETC’99, 1999 ASME International Design Engineering Technical Conferences, September 12-15, 1999, Nevada, Las Vegas, USA.

[8] U. Roy, B. Bharadwaj, A. Chavan, and C. K. Mohan. Development of a Feature Based Expert Manufacturing Process Planner. Proc. of the 7th IEEE International Conference on Tools with Artificial Intelligence, 1995, pp. 65-73.

[9] B. Bharadwaj. A Framework for Tolerance Synthesis of Mechanical Components. Master'' Thesis. Syracuse University, Syracuse, NY, 1995.